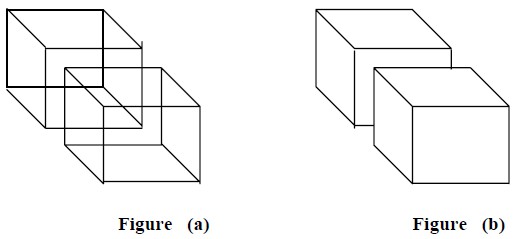
Unit 4 Visible Surface Determination



Visible Surface Determination

For the generation of realistic graphics display, we wish to determine which lines or surfaces of the   
objects are visible, either from the COP (for perspective projections) or along the direction of projection   
(for parallel projections), so that we can display only the visible lines or surfaces. For this, we need to   
conduct visibility tests. Visibility tests are conducted to determine the surface that is visible from a given   
viewpoint. This process is known as visible-line or visible-surface determination, or hidden-line or   
hidden-surface elimination. Visible surface detection or Hidden surface removal is major   
concern for realistic graphics for identifying those parts of a scene that are visible from a   
choosen viewing position.

Depending on the specified viewing position, particular edges are eliminated in the graphics display. For example Figure (a) represents more complex model and Figure (b) is a realistic view of the object, after removing hidden lines or edges.

There are numerous algorithms for identification of visible objects for different types of applications.

Some methods require more memory, some involve more processing time, and some apply only to special types of objects. Deciding upon a method for a particular application can depend on such factors as the complexity of the scene, type of objects to be displayed, available equipment, and whether static or   
animated displays are to be generated. These requirements have encouraged the development of carefully structured visible surface algorithms.

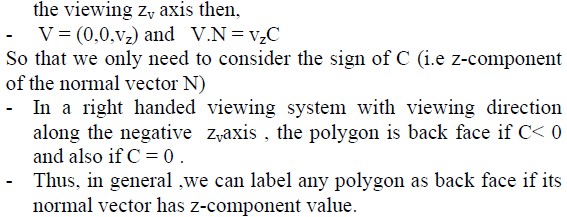
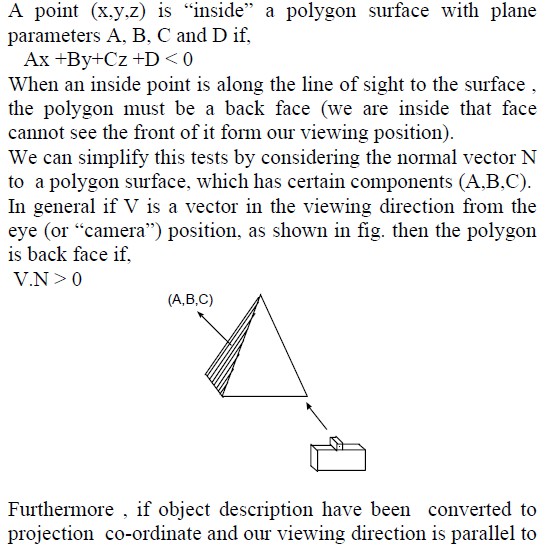
The two approaches are

Object-Space methods: An object space method compares objects and parts of objects to each other within scene definition to determine which surfaces are visible.

Image-Space methods: An image space method visibility is decided point by point at each pixel position on the projection plane.

Object space methods are implemented in the physical coordinate system in which objects are defined whereas image space methods are implemented in screen coordinate system in which the objects are viewed. Most visible surface algorithms use image-space method although object space method can be used effectively to locate visible surfaces in some cases. Line display algorithm generally use object space method to identify visible lines in wire-frame display.

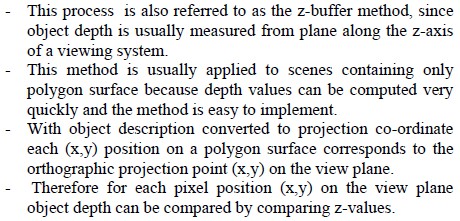
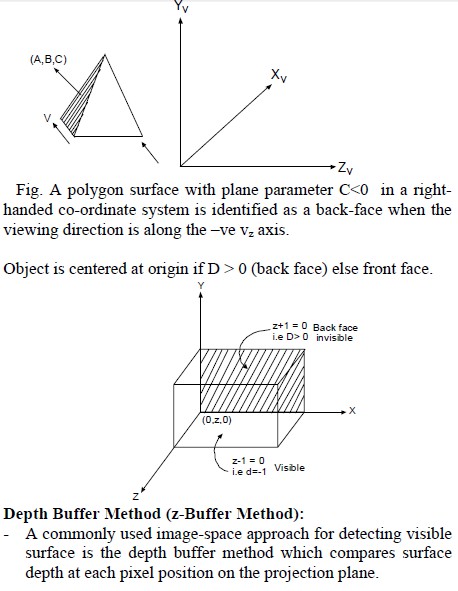
Unit 4 Visible Surface Determination



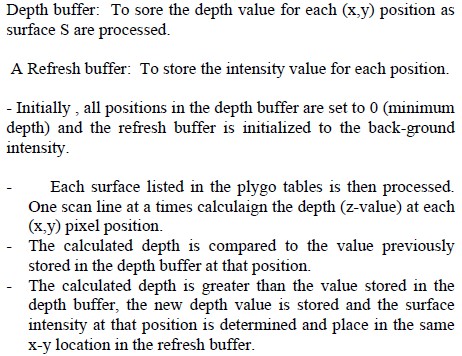
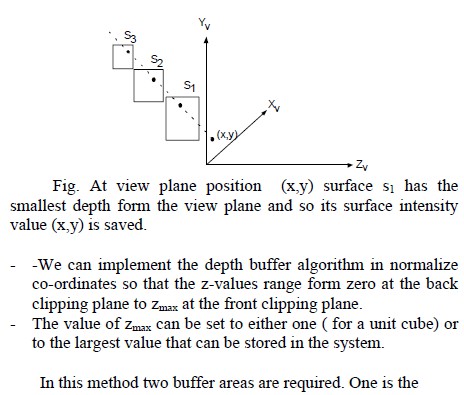
Back Face detection:

A fast and simple object space method for identifying the back-faces of a poly hedron based on the “inside-outside” tests.

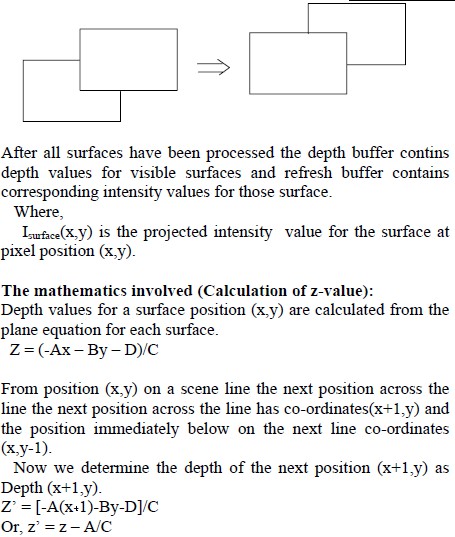
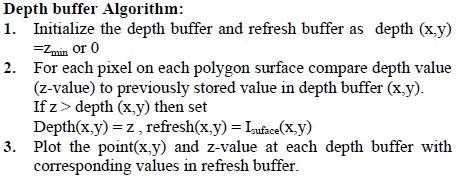
Unit 4 Visible Surface Determination



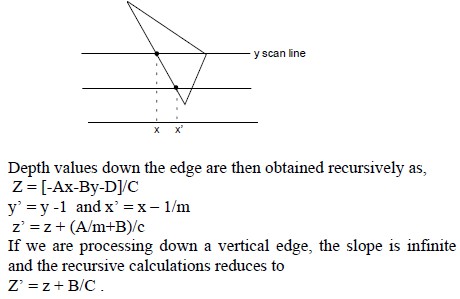
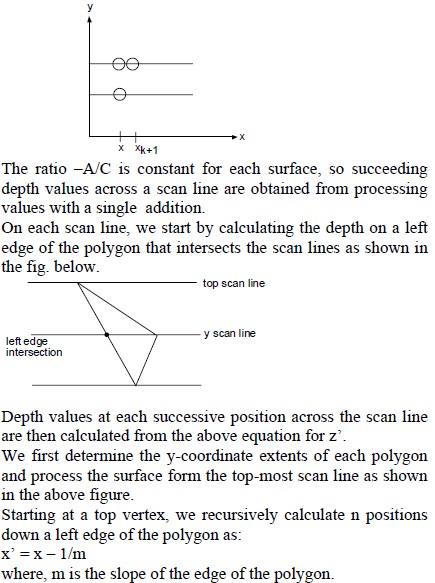
Unit 4 Visible Surface Determination



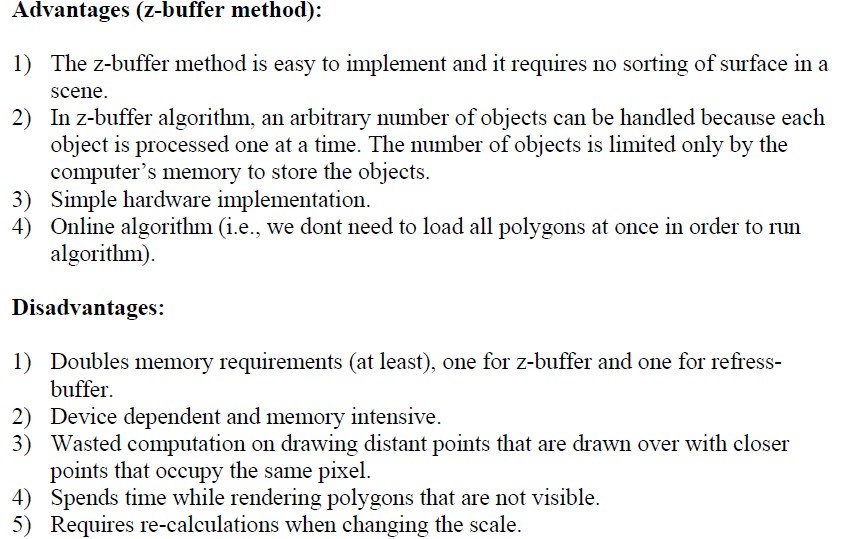
Unit 4 Visible Surface Determination



Unit 4 Visible Surface Determination



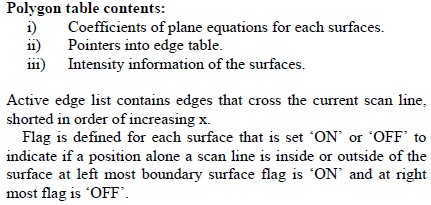
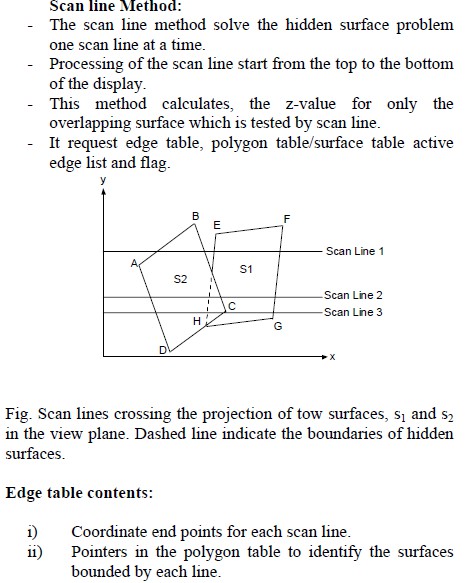
Unit 4 Visible Surface Determination



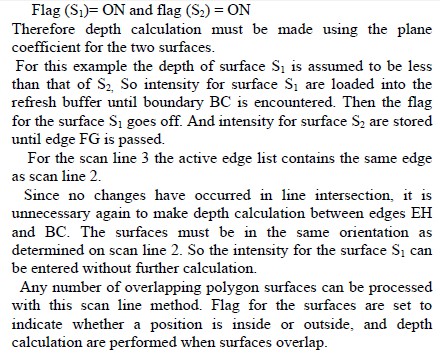
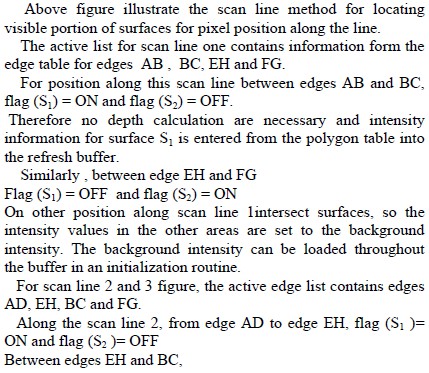
Scan-Line method

In contrast to z-buffer method, where we consider one surface at a time, scan-line method deals with multiple surfaces. As it processes each scan-line at a time, all polygon intersected by that scan-line are examined to determine which surfaces are visible. The visibility test involves the comparison of depths of each overlapping surface to determine which one is closer to the view plane. If it is found so, then it is declared as a visible surface and the intensity values at the positions along the scan-line are entered into the refresh-buffer.

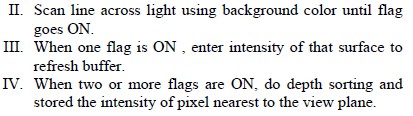
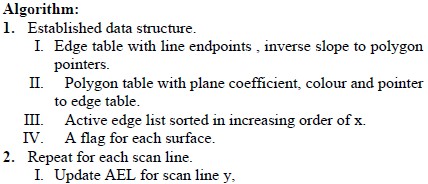
Unit 4 Visible Surface Determination



Unit 4 Visible Surface Determination



Unit 4 Visible Surface Determination



List Priority Algorithms

•Determine a visibility ordering for objects ensuring that a correct image results if the objects are rendered in that order

•Use a combination of object precision and image precision operations

•Object precision: depth comparisons and object splitting

•Image precision: scan conversion

•The list of sorted objects is created with object precision

•Two examples: Depth sort algorithm (Painter’s Algorithm and BSP’s)

DEPTH SORTING ALGORITHM:

This method uses both object space and image space method. In this method the surface

representation of 3D object are sorted in of decreasing depth from viewer. Then sorted surface are scan converted in order starting with surface of greatest depth for the viewer.   
The conceptual steps that performed in depth-sort algorithm are

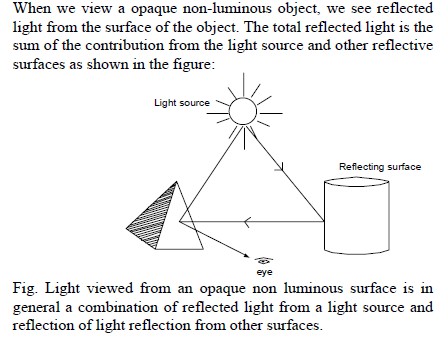
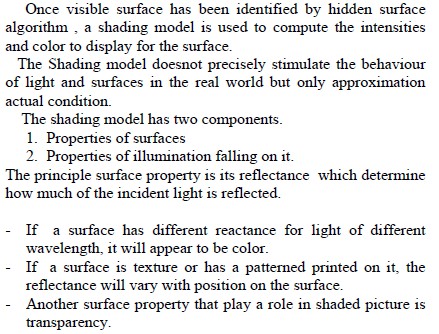
1. Sort all polygon surface according to the smallest (farthest) Z co-ordinate of each.

2. Resolve any ambiguity this may cause when the polygons Z extents overlap, splitting polygons if necessary.

3. Scan convert each polygon in ascending order of smaller Z-co-ordinate i.e. farthest surface first (back to front)

In this method, the newly displayed surface is partly or completely obscures the previously   
displayed surface. Essentially, we are sorting the surface into priority order such that surface   
with lower priority (lower z, far objects) can be obscured by those with higher priority (high z-  
value).

Unit 4 Visible Surface Determination

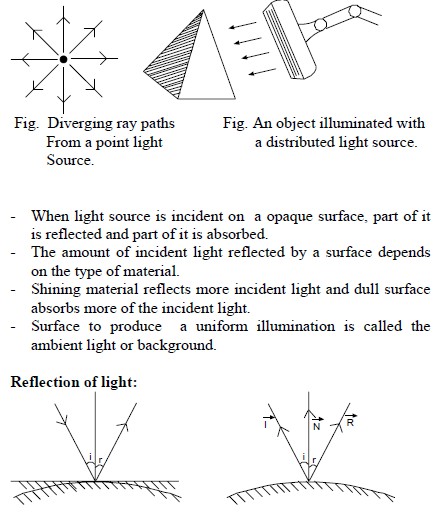
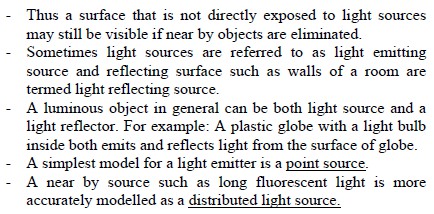


This algorithm is also called "Painter's Algorithm" as it simulates how a painter typically produces his painting by starting with the background and then progressively adding new (nearer) objects to the canvas.

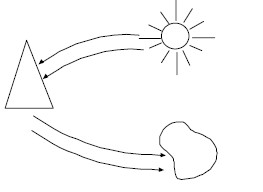
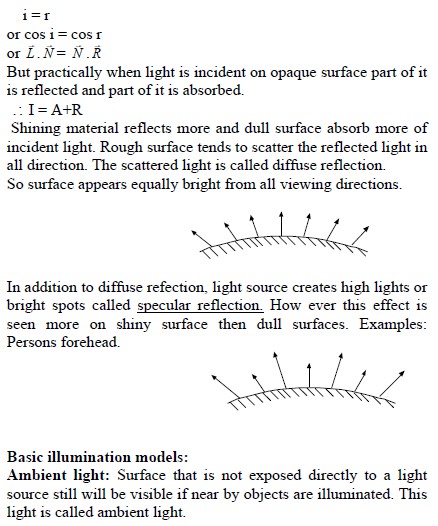
Illumination and Shading

Light Source

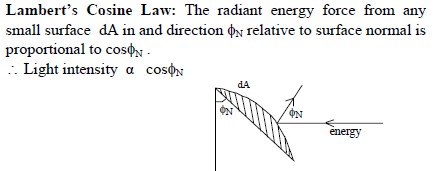
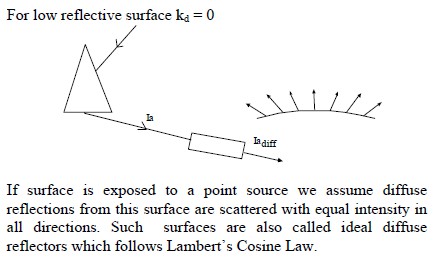
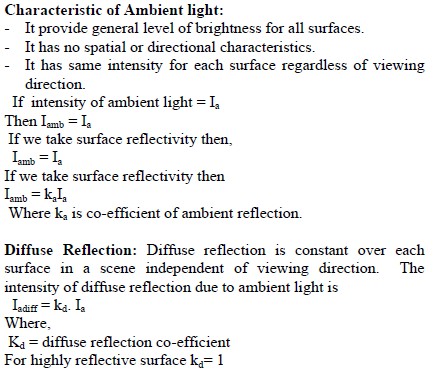
Unit 4 Visible Surface Determination



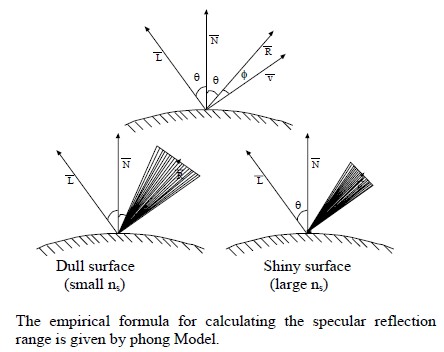
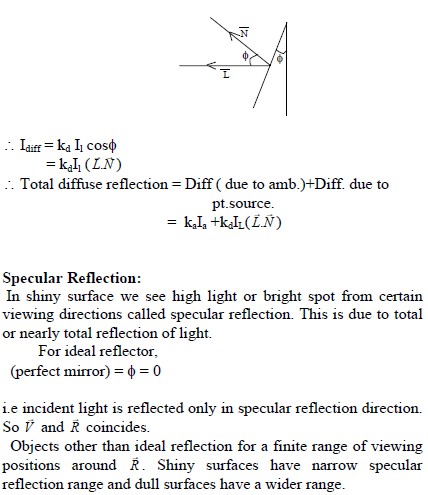
Unit 4 Visible Surface Determination



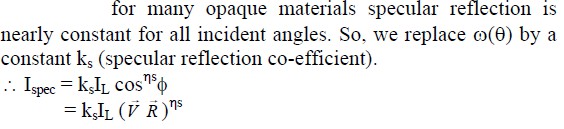
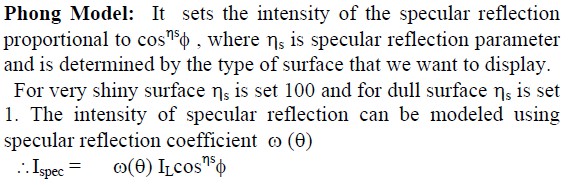
Unit 4 Visible Surface Determination



Unit 4 Visible Surface Determination



Unit 4 Visible Surface Determination



Polygon Rendering Methods:

(i) Constant Intensity shading Method.

(ii) Gouraud Shading method(Intensity Interpolation)

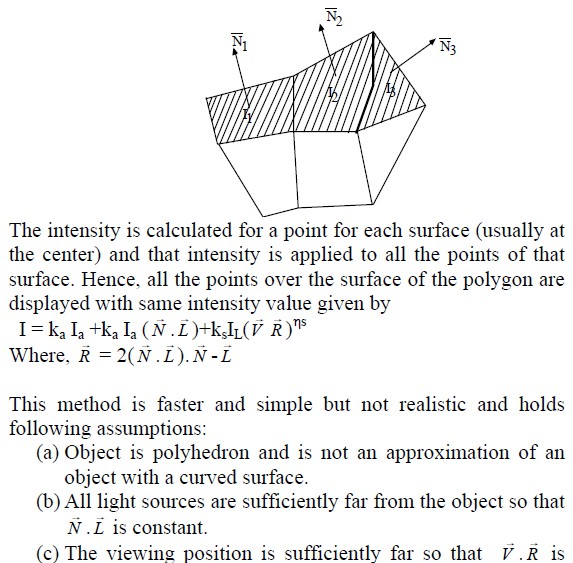
(iii) Phong Shading Method (Normal Vector Interpolation).

(i) Constant Intensity shading Method.

The simplest model for shading for a polygon is constant intensity shading also

called as Faceted Shading Or flat shading. This approach implies an illumination model once to determine a single intensity value that is then used to render an entire polygon. Constant shading is useful for quickly displaying the general   
appearance of a curved surface.

Unit 4 Visible Surface Determination



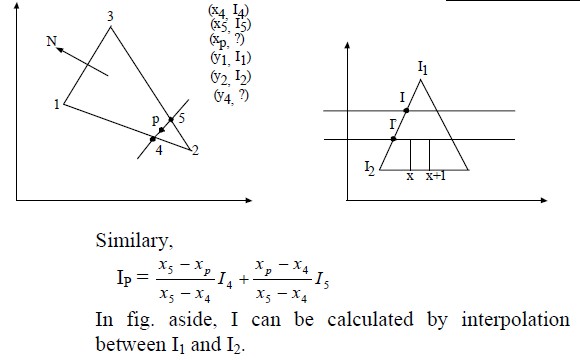
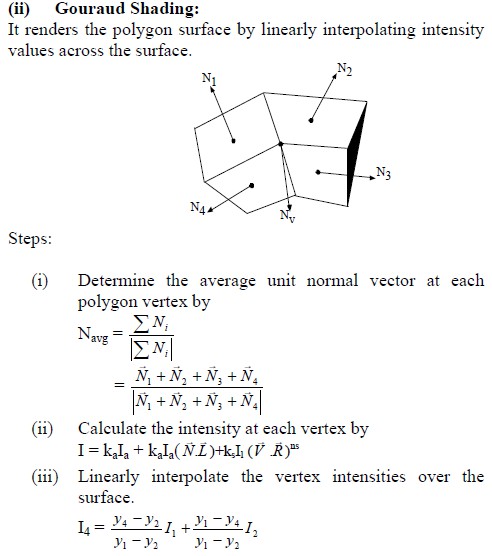
Constant.

Even if all conditions are not true, we can still reasonably approximate surface -  
lighting effects using small polygon facets with fast shading and calculate the   
intensity for each facet, at the centre of the polygon of course constant shading   
does not produce the variations in shade across the polygon that should occur.

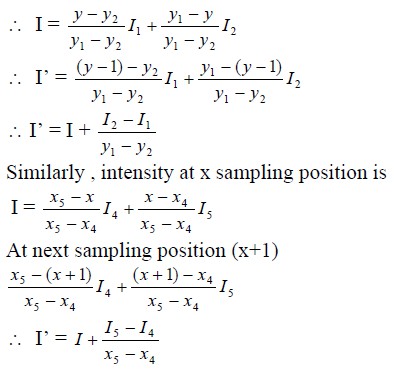
Disadvantage:

- The intensity discontinuity occurs at the border of the two surfaces.

Unit 4 Visible Surface Determination



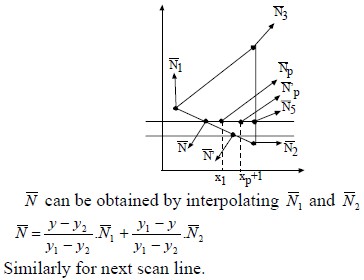
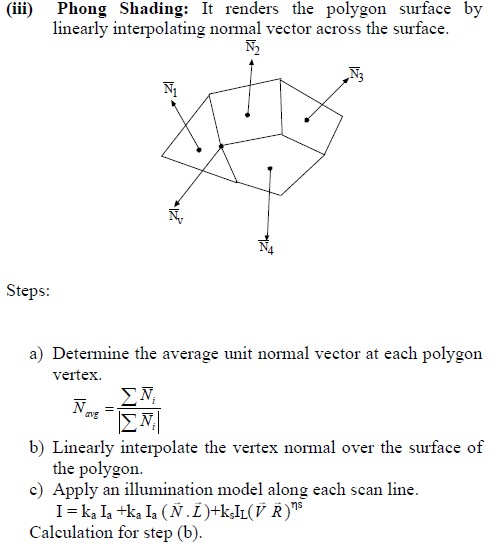
Unit 4 Visible Surface Determination



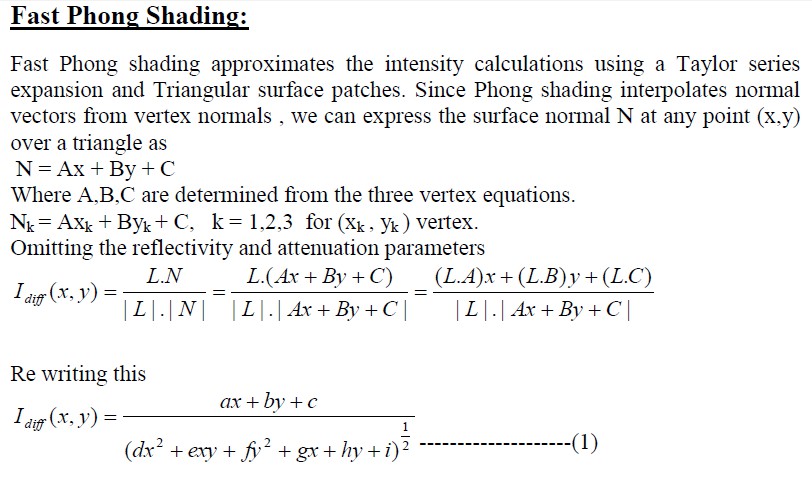
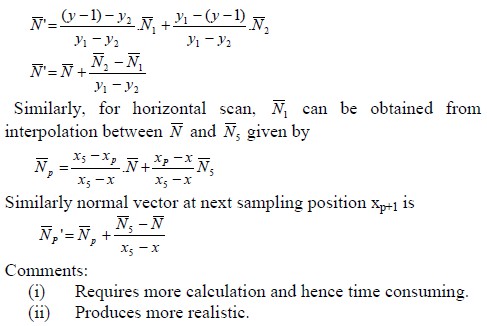
Phong Shading

A more accurate method for rendering a polygon surface is to interpolate normal   
vector and then apply illumination model to each surface point. This method is   
called Phong shading or normal vector interpolation method for shading. It   
displays more realistic highlights and greatly reduce the mach band effect.   
A polygon surface is rendered with Phong shading by carrying out following   
calculations.

Unit 4 Visible Surface Determination



Unit 4 Visible Surface Determination



Unit 4 Visible Surface Determination

